

Client einrichten

Beschreibung:

Um auf das git Repository Zuzugreifen brauchen wir den Namen und den Schlüssel genauso die URL.

Haben wir eine Neue Repo über die Weboberfläche erstellt und die ist noch leer. Dann können wir unser Verzeichnis lokal fürs Pushen vorbereiten. Ist da schon was drin wäre, es ein Pull.

Daten im git config Verzeichnis ändern:

```
git config user.email "meinepersonalemail"  
git config user.name "immohacker"
```

Man kann aber auch einfach im .git Verzeichnis die config Datei editieren und am ende folgendes hinzufügen, dann braucht man keine Befehle.

```
...  
[user]  
  name = <gitusername>  
  email = info@example.com
```

Remote via HTTPS

Verzeichnis vorbereiten für Pull:

```
git init  
git checkout -b main  
git add -A  
git commit -m "first commit"  
git remote add origin https://git.example.com/Strange-Hosting/lb-hetzner-floating-ip.git  
git push -u origin main
```

Danach wird nach dem Git Benutzernamen und Benutzerkennwort gefragt

Ausgabe:

```
estehendes Git-Repository in /home/duffy/strangehosting/loadbalancer/.git/ neuinitialisiert
Zu neuem Branch 'main' gewechselt
[main (Root-Commit) ff1323f] first commit
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 README.md
error: externes Repository origin existiert bereits.
Username for 'https://git.hacker-net.de':
```

Nach Eingabe wurden die Dateien gepusht:

```
Bestehendes Git-Repository in /home/duffy/strangehosting/loadbalancer/.git/ neuinitialisiert
Zu neuem Branch 'main' gewechselt
[main (Root-Commit) ff1323f] first commit
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 README.md
error: externes Repository origin existiert bereits.
Username for 'https://git.hacker-net.de': duffyduck
Password for 'https://duffyduck@git.hacker-net.de':
Objekte aufzählen: 3, fertig.
Zähle Objekte: 100% (3/3), fertig.
Schreibe Objekte: 100% (3/3), 210 Bytes | 210.00 KiB/s, fertig.
Gesamt 3 (Delta 0), Wiederverwendet 0 (Delta 0), Pack wiederverwendet 0
remote: . Processing 1 references
remote: Processed 1 references in total
To https://git.hacker-net.de/Strange-Hosting/lb-hetzner-floating-ip.git
 * [new branch]    main -> main
Branch 'main' folgt nun Remote-Branch 'main' von 'origin'.
```

Um die Repo zu aktualisieren (Pushen) werden diese Befehlebenötigt

```
git add -A
git commit -m "first commit"
git push -u origin main
```

Um die Repo zu aktualisieren (Pullen) werden diese Befehlebenötigt

```
git pull
```

Eine Repo Clonen wenn auf dem Computer noch nichts eingerichtet ist.
Neues Verzeichnis erstellen.

```
git clone <Repository-URL>  
git clone https://git.hacker-net.de/Strange-Hosting/lb-hetzner-floating-ip.git
```

Dieser Befehl erstellt ein neues Verzeichnis mit dem Namen des Repositories im aktuellen Arbeitsverzeichnis, initialisiert ein .git Verzeichnis darin, lädt alle Daten für das Repository herunter und checkt eine Arbeitskopie der neuesten Version aus.

Remote VIA SSH

Wenn die Verbindung via SSH aufgebaut werden soll muss erst mal ein Schlüssel auf dem Client erstellt werden:

```
ssh-keygen -t ed25519 -C "your_email@example.com"
```

Nun das Ziel angeben:

Zum Beispiel /home/<username>/.ssh/gitea

```
Generating public/private ed25519 key pair.  
Enter file in which to save the key (/home/<username>/.ssh/id_ed25519): /home/<username>/.ssh/gitea
```

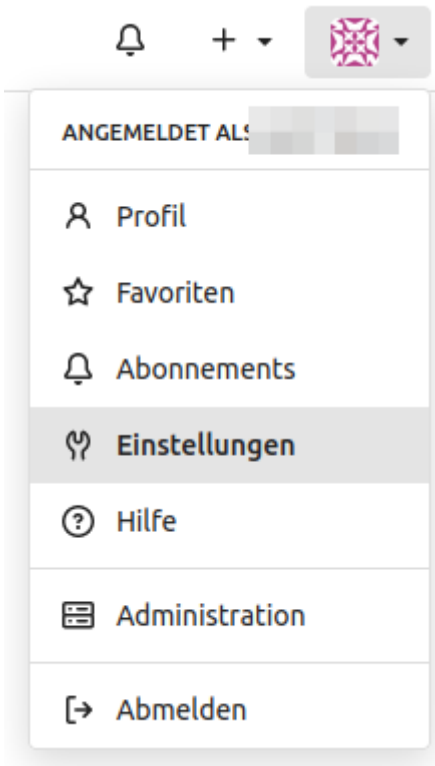
Wenn nach einer gefragt wird, leer lassen. Einfach nur enter / enter

```
Enter passphrase (empty for no passphrase):  
Enter same passphrase again:  
Your identification has been saved in /home/<username>/.ssh/gitea  
Your public key has been saved in /home/<username>/.ssh/gitea.pub  
The key fingerprint is:  
....
```

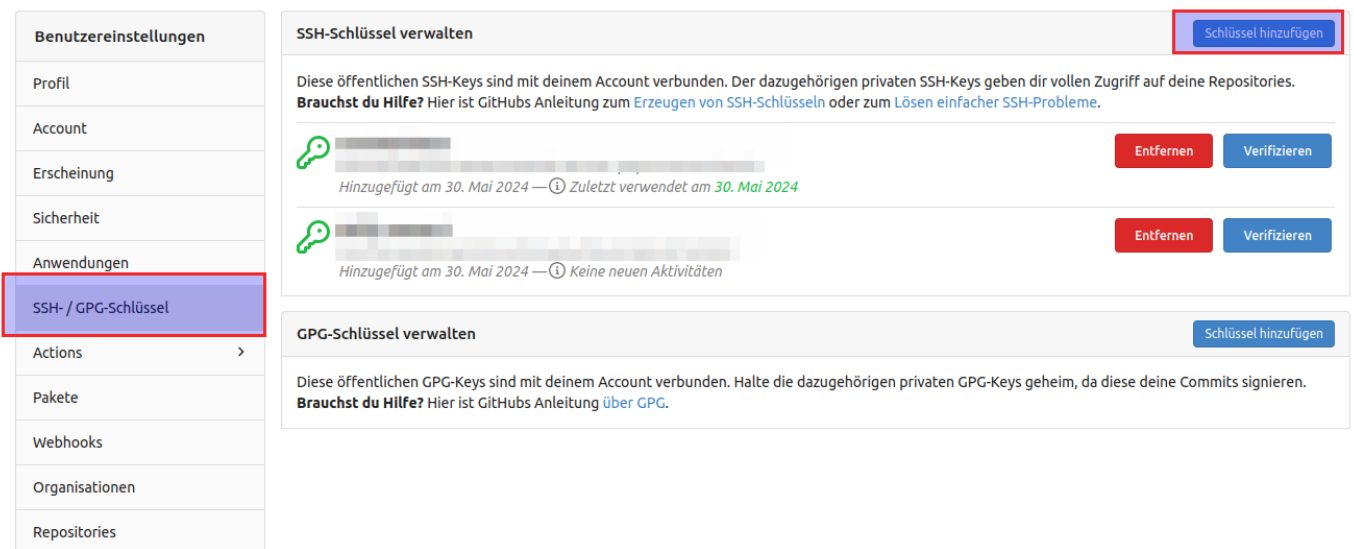
Den öffentlichen schlüssel anzeigen lassen und im git einfügen.
diesen kopieren

```
cat /home/<username>/.ssh/gitea.pub
```

Dazu gehen wir in Gitea unter Einstellungen



Dann unter SSH-GPG Schlüssel und auf Schlüssel hinzufügen klicken



Schlüsselname eintragen den Key einfügen und auf Schlüssel hinzufügen klicken.

SSH-Schlüssel verwalten

Schlüssel hinzufügen

Schlüsselname

Inhalt

```
startet mit 'ssh-ed25519', 'ssh-rsa', 'ecdsa-sha2-nistp256', 'ecdsa-sha2-nistp384', 'ecdsa-sha2-nistp521', 'sk-ecdsa-sha2-nistp256@openssh.com', oder 'sk-ssh-ed25519@openssh.com'
```

Schlüssel hinzufügen Abbrechen

Diese öffentlichen SSH-Keys sind mit deinem Account verbunden. Der dazugehörigen privaten SSH-Keys geben dir vollen Zugriff auf deine Repositories.
Brauchst du Hilfe? Hier ist GitHubs Anleitung zum [Erzeugen von SSH-Schlüsseln](#) oder zum [Lösen einfacher SSH-Probleme](#).

.ssh/config anpassen

Damit auch der Schlüssel für den Hostnamen genommen wird muss die config angepasst / erstellt werden

```
nano /home/<username>/.ssh/config
```

Dort einfügen / hinzufügen. Die IdentityFile ist der private ssh key

```
Host git.example.com
HostName git.example.com
User <deingitbenutzername>
IdentityFile ~/.ssh/gitea
```

Sollte der SSH Port ein anderer sein, als vom Server. Zum Beispiel weil gitea in einem Container läuft und der port ssh 222 von außen erreichbar ist muss in der ssh conf beim Client noch der Port hinzugefügt werden.

```
Host git.example.com
HostName git.example.com
User <deingitbenutzername>
Port 222
IdentityFile ~/.ssh/gitea
```

Nun die Rechte der Datei anpassen

```
chmod 600 ~/.ssh/config
```

Fertig nun kann sich per ssh verbunden werden.

zum schluss muss nur noch die URL geändert werden wenn vorher https war.

```
git remote remove origin  
git remote add origin git@git.example.com:Strange-Hosting/lb-hetzner-floating-ip.git
```

ansonsten nur

```
git remote add origin git@git.example.com:Strange-Hosting/lb-hetzner-floating-ip.git
```

oder in der .git/config datei die URL austauschen wenn vorher schon https war

```
...  
[remote "origin"]  
  url = git@git.example.com:Strange-Hosting/lb-hetzner-floating-ip.git  
...
```

Version #9

Erstellt: 24 März 2024 16:30:10 von Admin

Zuletzt aktualisiert: 30 Mai 2024 12:40:26 von Admin